```
  int stripe = Integer.parseInt(conf.get("stripe");
String line = value.tostring()
String matrix[] = value.split('s')

If (matrix[0],euqals("vector")

Do
         For (int a = 0 ; a< stripe;a++){
                  //first set key, to be row, stripe
                        Outputkey.set(matrix[1] + ',' + a);

                      //then set value, set the value to be vector, position, value.


                                   Outputvalue.set(matrix[0]+","+matrix[2]+","+matrix[3];
                                   Context.write(key, value)



If matrix[0] equals("matrix")
Do
                  For (int a = 0 ; a< stripe;a++){



// set the key to be stripe, row.
                          Outputkey.set(a + "," + matrix[2]);

                    Outputvalue.set (matrix[0]+ "," + matrix[1]+matrix[3])
Context.write    (key, value)
}




Reducer:
Hashmap a     for matrix data
Hashmap b     for vector data

for (val:values)
{

If it is from matrix{
Put the (key, value) in hashmap a

}
```

Else put to hashmap b.

Multiplication process:

Mapper : identity mapper is sufficient since we are already splitting the values up into stripes and put the key value pairs inside the hashmap.

Reducer:
Hashmap a , hashmap b .

Result;

Get the m_ij value from hashmap a and n_jk value from hashmap b and add result by mij * njk

427 HW7 PG1

Q1

int s = Integer.parse

Δ stripping process.

input. $\begin{bmatrix} 16 & 2 & 3 & 13 \\ 5 & 11 & 10 & 8 \\ 9 & 7 & 6 & 12 \\ 4 & 14 & 15 & 1 \end{bmatrix}$ $\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$

output. $\{(\underset{\text{strip}}{1}, \underset{\text{row}}{1}), (m, 1, \underset{\text{position}}{16})\}$

$\{(1,1), (m,2,2)\}$

$\{(1,2), (m,1,5)\}$

$\{(1,2), (m,2,11)\}$

$\{(1,3), (m,1,9)\}$

$\{(1,3), (m,2,7)\}$

$\{(1,4), (m,1,4)\}$

$\{(1,4), (m,2,14)\}$

$\{(2,1), (m,1,3)\}$

$\{(2,1), (m,2,13)\}$

$\{(2,2), (m,1,10)\}$

427 HW7 PG2

$\{(2,2), (m,2,8)\}$

$\{(2,3), (m,1,6)\}$

$\{(2,3), (m,2,12)\}$

$\{(2,4), (m,1,15)\}$

$\{(2,4), (m,1,1)\}$

$\Rightarrow \{(1,1), (V,1,1)$

row stripe

$\{(2,1)\ (V,1,2)\}$

$\{(1,2)\ (V,1,3)\}$

$\{(2,2)\ (V,1,4)\}$

Q1 Part c ,

2.5.1 a

The communication cost for matrix multiplication is O(row*column)

2.5.1 b

The communication cost for union two matrix R and S is O(number of entries in R + number of entries in S)

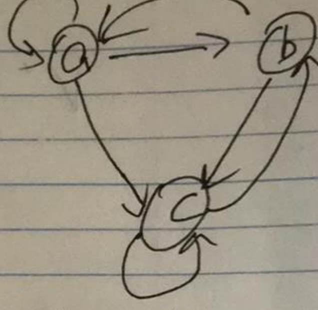
2.5..1 C

The communication cost is the number of tuples in the relation R.

Q2

Q2 - part 1.

$$\begin{pmatrix} 0.23076923 \\ 0.3076923 \\ 0.46153846 \end{pmatrix}$$



the transit matrix M is
$$\begin{bmatrix} \frac{1}{3} & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

$$V = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}$$

$$(M)^{100} \times V = \begin{pmatrix} 0.23077 \\ 0.30769 \\ 0.46154 \end{pmatrix}$$

Q2 part 2.

$$V' = \beta M V + \underbrace{(1-\beta) e / n}_{\text{always constant}}$$

$$= 0.2 \times \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} / 3.$$

$$= \begin{bmatrix} \frac{1}{15} \\ \frac{1}{15} \\ \frac{1}{15} \end{bmatrix} \Rightarrow \text{set it to be } K$$

$$V^{n+1} = \left[(\beta M)^n + (\beta M)^{n-1} + \cdots + (\beta M)^1 \right] K + M^{n+1} V$$

$$= \begin{pmatrix} 0.19259 \\ 0.24198 \\ 0.36543 \end{pmatrix} + \begin{pmatrix} 0.23077 \\ 0.30769 \\ 0.46154 \end{pmatrix}$$

$$= \begin{pmatrix} 0.42336 \\ 0.54967 \\ 0.82697 \end{pmatrix}.$$

Q2 part 3

① Figure 5.4.

Source	Degree	Destinations
A	3.	B C D
B	2	A. D.
C	1	E
D	2.	B C.
E.	0	—

Figure 5.7

Source	Degree	Destination
a	3	a, b, c.
b	2	a, c.
c	2	b, c.

This is more effecient because the transition matrix is generally very sparse, which mean there could possibly be tons of zeros. So by only representing no zero values reduce the amount of data thus improve effiency.

Q3 part a .

it has y different keys, for each key, there are x+z key-value pairs, There are xy+xz key value pairs in total.

Part b

For the first reducer phase,

Because we need to calculate each possible combination of row column pairs, so there are xz possible keys,

The length of the value is y

Part c

The output of second map phase has xz number of keys, because it is an identity mapper. And there are y value pairs associated with each key. So there are xyz pairs in total.

Part d The usage cost is xyz+xy+xz.

The communication cost is xy.

e. partitioning in to block size will increase the communication cost, because it will now increase the communication between blocks, and decreases the usage cost.

The optimal block size should be (x+z)/y